พัฒนาการภาษาไทย
บนเครื่องไมโครคอมพิวเตอร์
ตอน ๑
(TKS-Thai Kernel System)

ยืน ภู่วรวรรณ
พรพจน์ สุริยะวงศ์

บทนำ

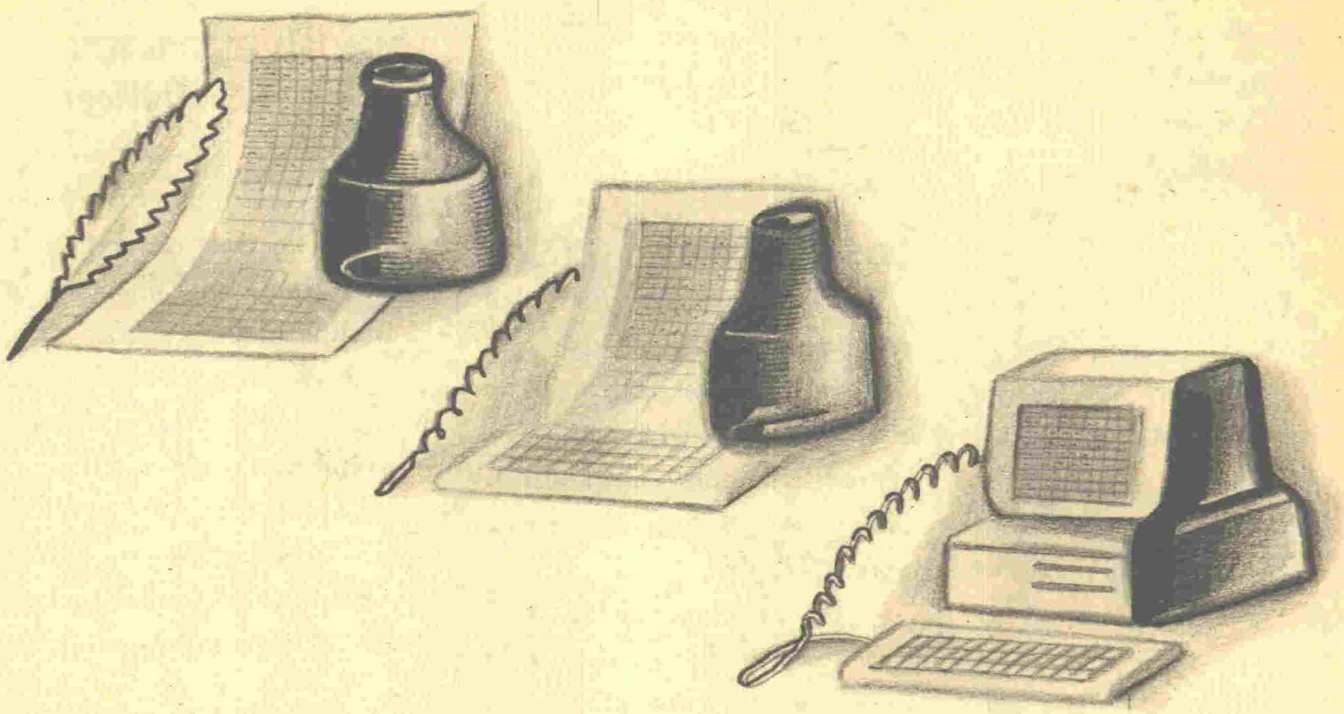
พัฒนาการภาษาไทยบนเครื่องคอมพิวเตอร์มีความเป็นมา ในลักษณะที่ก้าวหน้าเป็นขั้นตอนตั้งแต่ครั้งเริ่มใช้งานคอมพิวเตอร์ หากจะแบ่งยุคของการพัฒนาภาษาไทยบนเครื่องไมโครคอมพิวเตอร์ ผู้เขียนขอแบ่งยุคของระบบภาษาไทยออกได้เป็นดังนี้

ยุคที่ 1 เป็นยุคของการประมวลผลภาษาไทยยุคแรก เป็นยุคที่ผู้พัฒนาภาษาไทยจะทำบนเมนเฟรมโดยเขียนโปรแกรมเป็นโปรแกรมย่อยในระดับภาษาชั้นสูง เช่น โคบอล หรือฟอร์แทรน และถ้าต้องการประมวลผลเช่นการจัดระดับการเรียงพิมพ์ก็จะเรียกมายังโปรแกรมย่อย งานที่ทำส่วนใหญ่จึงเป็นงานลักษณะแบตช์ ดังนั้นจึงไม่มีปัญหามากนัก เพราะเรายังไม่ต้องการการตอบโต้กับคอมพิวเตอร์แบบ interactive งานประมวลผลในยุคนั้นจึงเขียนกันเป็นรูทีน แล้วคอมพายล์ด้วยภาษาระดับสูงให้ผู้เขียนโปรแกรมลิงค์ใช้งานต่อไป ตัวอย่างของการประมวลผลในที่นี้จะพบเห็นในระดับเครื่องคอมพิวเตอร์เมนเฟรมในยุค IBM 360, IBM 370 เป็นต้น

ยุคที่ 2 เป็นยุคของการประมวลผลภาษาไทยบนเครื่องไมโครคอมพิวเตอร์ช่วงแรก ตั้งแต่เครื่อง 8 บิตเป็นต้นมา การประมวลผลภาษาไทยเน้นการแก้ไขเปลี่ยนแปลงตัวกำเนิดอักษร (Character generator) แล้วสร้างจอภาพของการแสดงผลแบบ 8 บรรทัด การเปลี่ยนแปลงฮาร์ดแวร์ยังไม่มีอะไรมากเพียงเพิ่มเติม ROM กำเนิดตัวอักษร การพัฒนาทางซอฟต์แวร์เน้นลักษณะของการสร้างโปรแกรมไดรเวอร์ที่แฝงลงไปทำงานร่วมกับโอเอส มีการเขียนเวิร์ดโปรเซสเซอร์ และพัฒนาให้โปรแกรมสำเร็จรูปรับภาษาไทยในระบบ 8 บิตได้ แต่การจัดสระบางครั้งยังไม่สามารถทำได้ดี โปรแกรมที่ทำได้เช่น ดีเบสทู ซูเปอร์แคลค์ ฯลฯ การเปลี่ยนแปลงเครื่องพิมพ์ก็สามารถทำได้ดีขึ้นคือเครื่องพิมพ์ดอตแมตริกซ์สามารถพิมพ์อักษรไทยได้ดี

ยุคที่ 3 เป็นยุคที่มีการแข่งขันในเรื่องไทยการ์ด มีการพัฒนาปรับปรุงโครงสร้างการแสดงผลด้วยเทคนิคต่าง ๆ กันเพื่อให้แสดงผลได้ 25 บรรทัด โครงสร้างของการปรับปรุงทางฮาร์ดแวร์จะจัดทำขึ้นเป็นพิเศษ มีการเขียนโปรแกรมไดรเวอร์ในรูปแบบต่าง ๆ กันเพื่อให้ใช้ได้กับโปรแกรมสำเร็จรูปหลาย ๆ โปรแกรมได้ การใช้งานภาษาไทยในยุคนี้ถึงแม้ว่าตัวอักษรจะเล็กไปสักนิดแต่ก็ใช้งานได้ผลดี เพราะสามารถใช้ได้กับโปรแกรมสำเร็จรูปเช่น ดีเบส โลตัส ได้ อย่างไรก็ตาม การพัฒนาทางซอฟต์แวร์ในระดับโปรแกรมสำเร็จรูปก็ยังคงยึดโปรแกรมเวิร์ดโปรเซสเซอร์เป็นสำคัญ

ยุคที่ 4 เป็นยุคที่ผู้เขียนมีความคิดเห็นว่าระบบภาษาไทยที่ใช้กันนี้น่าจะมีการพัฒนาไปในลู่ทางของการแก้ปัญหาในยุคที่ 3 ได้ คือระบบซอฟต์แวร์ที่ทำงานอยู่น่าจะวิ่งบนไทยการ์ดได้ทุกรุ่นทุกแบบ โครงสร้างภาษาไทยที่ใช้งานน่าจะทำให้ผู้ใช้ได้ประโยชน์มากที่สุด ระบบจึงไม่ควรผูกไว้กับฮาร์ดแวร์ นอกจากนี้ยุคของการแสดงผลทั้งทางด้านจอภาพและเครื่องพิมพ์ได้ผ่านยุคสมัยจาก low resolution เป็น high resolution ดังจะเห็นได้ว่าเรามีจอภาพ EGA VGA ที่ให้รายละเอียดสูง ซึ่งทำให้แสดงผลภาษาไทยทำได้ดีขึ้น ประจวบกับซีพียูมีความเร็วสูงขึ้น การพัฒนาภาษาไทยจะเปลี่ยนแปลงไปตามเทคโนโลยี ในยุคนี้ระบบภาษาไทยน่าจะหันเข้าหามาตรฐาน



กลางที่ใคร่อยากเสนอแนะต่อไป เพื่อว่าซอฟต์แวร์ที่พัฒนาขึ้นจะได้เป็นอิสระต่อฮาร์ดแวร์ได้

ทำไมต้องมีระบบภาษาไทยระดับแก่น

ปัญหาของการพัฒนาโปรแกรมประยุกต์ภาษาไทยที่พบเห็นกันอยู่ในขณะนี้ คือซอฟต์แวร์ที่พัฒนาขึ้นมานั้นมักผูกพันกับระบบภาษาไทยระบบใดระบบหนึ่งโดยเฉพาะ เช่น โปรแกรมเวิร์ดโปรเซสเซอร์ภาษาไทยที่เขียนขึ้นจะใช้ได้กับระบบไทยการ์ดของบริษัทใดบริษัทหนึ่ง ไม่สามารถนำมาใช้กับระบบที่แตกต่างกันได้ งานพัฒนาซอฟต์แวร์ดังกล่าวจึงมีข้อจำกัดที่ทำให้ไม่สามารถใช้ได้อย่างกว้างขวางกับทุกระบบที่มีขายอยู่ในท้องตลาด

หากสังเกตการพัฒนาซอฟต์แวร์ระบบไมโครคอมพิวเตอร์ในต่างประเทศจะพบว่าผู้พัฒนาจะให้ระบบซอฟต์แวร์ที่ขึ้นกับฮาร์ดแวร์อยู่ในระดับต่ำสุด เช่น โปรแกรมไบออส BIOS-Basic Input Output System และกำหนดการเรียกเข้าหาระบบอย่างมีแบบแผน จึงทำให้โครงสร้างของไมโครคอมพิวเตอร์ที่มีฮาร์ดแวร์หลายรูปแบบ ใช้งานกับซอฟต์แวร์ร่วมกันได้ เช่น โมดูลของไบออสที่จัดการเกี่ยวกับจอสีจอเขียว หรือจอภาพที่มีรายละเอียดแตกต่างกันก็สามารถใช้งานร่วมกันได้ การกำหนดรูปแบบของซอฟต์แวร์จึงกำหนดเป็นระดับตั้งแต่โครงสร้างระดับแก่นออกมาที่ผิว

ในการพัฒนาโปรแกรมประยุกต์ภาษาไทยก็เช่นกัน ไม่ว่าเราจะประยุกต์ใช้กับไมโครคอมพิวเตอร์รุ่นใด มินิ หรือเมนเฟรมคอมพิวเตอร์ก็ตาม สิ่งที่เป็นพื้นฐานของการพัฒนาจะเหมือนกัน โดยเฉพาะอย่างยิ่งในส่วนที่เกี่ยวข้องกับอินพุตและเอาต์พุตภาษาไทย เช่น การแสดงผล การพิมพ์ผลลัพธ์ และการรับคีย์บอร์ด อย่างไรก็ดีถ้าให้ผู้เขียนซอฟต์แวร์หลาย ๆ คนเขียนโปรแกรมประยุกต์กับงานทางภาษาไทย จะพบว่าต่างคนก็จะเขียนไปคนละแบบอย่าง แต่จุดมุ่งหมายก็แสดงผลได้เหมือนกัน ดังนั้นหากมีการกำหนดรูปแบบโมดูลเหล่านี้ไว้ชัดเจนว่าเป็นฟังก์ชันอะไร มีการเรียกเข้าหาหรือใช้ฟังก์ชันได้อย่างไร มีตัวแปรอะไรเป็นอินพุตและเอาต์พุต รูปแบบที่กำหนดนี้ก็จะเป็นมาตรฐานกลางได้ โดยโปรแกรมที่เรียกหาฟังก์ชันเหล่านี้จะใช้เหมือนกันไม่ว่าจะอยู่บนระบบไมโครคอมพิวเตอร์อะไร การเรียกหาในระดับสูงขึ้นไปก็จะเกิดการเขียนโปรแกรมในลักษณะอย่างมีหลักการ กล่าวคือผู้เขียนซอฟต์แวร์ระดับสูงจะเขียนได้ง่ายขึ้น มีโครงสร้างเป็นมาตรฐานที่ไม่ผูกพันเฉพาะเจาะจงกับฮาร์ดแวร์

เมื่อเป็นเช่นนี้การพัฒนาระบบภาษาไทยบนไมโครคอมพิวเตอร์ก็น่าจะทำในลักษณะตามหลักการที่กล่าวมา กล่าวคือไม่ว่าเทคโนโลยีจะเปลี่ยนแปลงไปเช่นไร จอภาพที่ใช้จะเป็นสีหรือจอเขียว มีความละเอียดแตกต่างกัน รหัสภาษาไทยที่ใช้อาจแตกต่างกัน เครื่องพิมพ์ก็ต่างชนิด โปรแกรมจัดการระบบภาษาไทยในระดับต่ำก็จะจัดการให้ได้ โดยผู้เขียนประยุกต์ใช้งานจะไม่พบปัญหา

จุดมุ่งหมายของการจัดสร้างระบบภาษาไทยระดับแก่น

ระบบภาษาไทยระดับแก่น-Thai Kernel System ซึ่งต่อไปนี้จะเรียกว่า TKS เป็นระบบที่พัฒนาขึ้นมาเพื่อแก้ปัญหา

ลักษณะของความไม่ compatible กันในระดับการใช้ภาษาไทยในไมโครคอมพิวเตอร์

TKS เป็นฟังก์ชันบริการที่จะจัดการด้านภาษาไทยให้กับผู้เรียก โครงสร้างของ TKS ที่เสนอแนะนี้ได้รับการออกแบบโดยมีจุดมุ่งหมายที่สำคัญ ดังนี้

1. มีลักษณะเป็นฟังก์ชันบริการที่ง่ายต่อการเรียกใช้ แต่ละฟังก์ชันบริการจะมีหมายเลขกำกับ การเรียกใช้จะเรียกในลักษณะฟังก์ชัน CALL จากโปรแกรมประยุกต์

2. สนับสนุนการทำงานพื้นฐานที่เกี่ยวข้องกับภาษาไทยไว้ทั้งหมด เช่น ฟังก์ชันเกี่ยวกับจอภาพ คีย์บอร์ด เครื่องพิมพ์

3. มีความอ่อนตัวที่จะเปลี่ยนแปลงรูปแบบภายในให้ขึ้นกับอุปกรณ์ต่าง ๆ ได้ เช่นเครื่องพิมพ์ทุกชนิด การแสดงผลบนจอภาพได้ทุกโหมด (โหมดตัวอักษรหรือกราฟิก) รหัสการแทนค่าภาษาไทยทำได้ทุกรหัสและทุกระดับ ฯลฯ

4. TKS สามารถเพิ่มขยายต่อฟังก์ชันได้อีกตามความจำเป็น เช่น การจัดการจอภาพแบบช่องหน้าต่าง

5. เป็นส่วนเดียวของระบบซอฟต์แวร์ที่ยอมให้ผูกพันกับสิ่งแวดล้อมและฮาร์ดแวร์ แต่จะมีการกำหนดหน้าที่ที่แน่นอนเพื่อให้ผู้เขียน TKS ในโอกาสต่อไปเขียนได้ถูกต้องเพื่อสร้างความ compatibility

โครงสร้างของ TKS

เนื่องจาก TKS เป็นโครงสร้างที่อยู่ระดับชิดกับฮาร์ดแวร์และเป็นระดับต่ำสุดที่เกี่ยวข้องกับการจัดการภาษาไทย โครงสร้างของ TKS จึงมีการจัดวางตัวเองเพื่อให้การเรียกใช้เป็นไปอย่างมีประสิทธิภาพ โครงสร้างของ TKS เมื่อเขียนเป็นโครงสร้างจะแสดงได้ดังรูปที่ 1

โครงสร้างของระบบภาษาไทยที่เสนอในที่นี้จะมองเห็นฮาร์ดแวร์อยู่ในระดับต่ำสุด ไทยการ์ดทุกแบบหรือวิธีการแสดงผลทุกเทคโนโลยีไม่ว่าจะเป็นการใช้ฮาร์ดแวร์ช่วยหรือแบบบิตแมพจะอยู่ในระดับต่ำสุด ดังนั้นฮาร์ดแวร์จะเป็นอะไรก็ได้ แต่ถูกควบคุมด้วยซอฟต์แวร์ที่ควบคุมนี้เรียกว่าฟิสิคัลดีไวซ์ไดรเวอร์ และถัดขึ้นมาคือส่วนของลอจิคอลดีไวซ์ไดรเวอร์ และระดับก่อนถึงโปรแกรมประยุกต์อีกส่วนหนึ่ง คือฟังก์ชันคอลและไลบรารี่ ในที่นี้จะขอขยายความแต่ละส่วนต่อไป

ฟิสิคัลดีไวซ์ไดรเวอร์ (PDD-Physical Device Driver)

เป็นส่วนที่ขึ้นกับอุปกรณ์ที่ใช้แต่ละชนิด แต่จะมีลักษณะการเรียกใช้งานเป็นหมายเลข โดยกำหนดหน้าที่ที่แน่นอน เพื่อจะได้ทำให้มีความเป็นอิสระของระดับที่อยู่สูงกว่าส่วนของ PDD จะติดต่อกับอุปกรณ์ทางฮาร์ดแวร์ เช่น จอภาพ คีย์บอร์ด เครื่องพิมพ์ จานแม่เหล็ก ฯลฯ

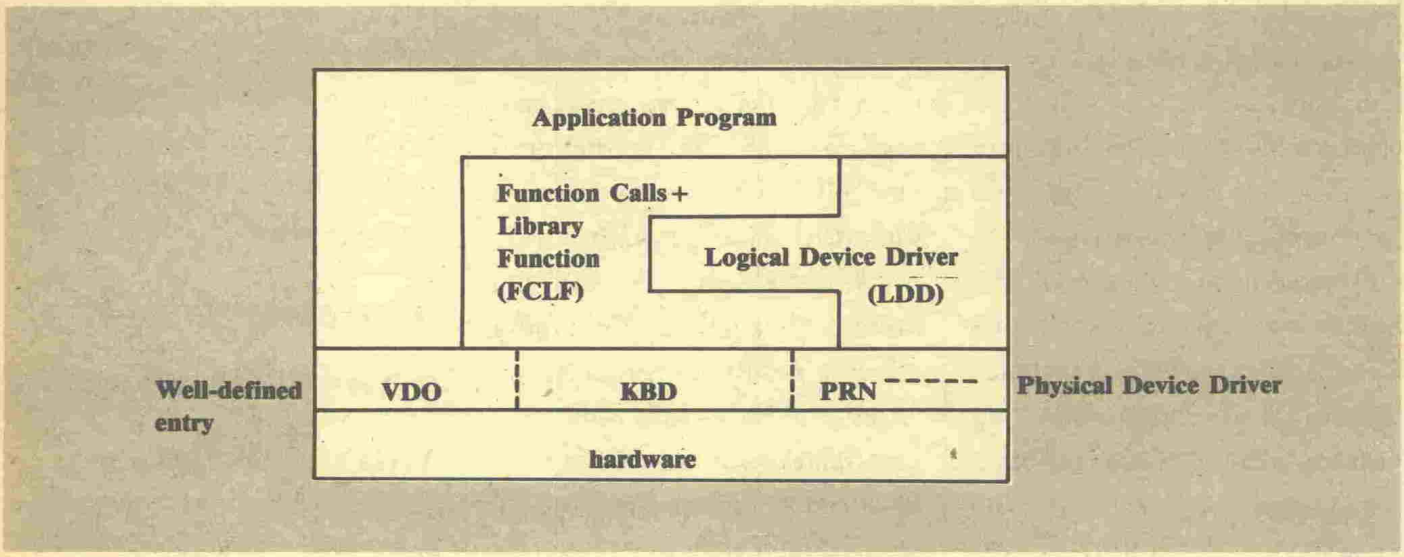
โครงสร้างภายในของ PDD มีลักษณะเป็น well-defined entry คือมีการกำหนดเป็นหมายเลขที่ชัดเจนเพื่อประโยชน์การเรียกใช้ หมายเลขที่กำหนดไว้จะเป็นดังนี้

entry #0-#3 มีหน้าที่เกี่ยวกับการตรวจสอบและการติดตั้งระบบ

entry #16-#21 มีหน้าที่เกี่ยวกับการรับและส่งข้อมูลเป็นตัวอักษร (char) หรือเป็นคำ (word)

entry #22-#25 มีหน้าที่เกี่ยวกับการรับและส่งข้อมูลเป็นกลุ่มหรือสตรีม (stream)

ตัวอย่างของหมายเลขที่เรียกเข้าหาระบบบางตัวแสดงได้ดังตารางที่ 1



รูปที่ 1 โครงสร้างส่วนประกอบของ TKS

สิ่งที่น่าสังเกตคือ การส่งตัวอักษรในรูปแบบตัวอักษรหรือเวิร์ดนั้นจะใช้รหัสอะไรก็ได้ และสำหรับรหัสภาษาไทยที่ใช้จะเป็นรหัสแบบมีไบต์เดียวหรือสองไบต์ก็ได้ ทำนองเดียวกันกับโครงสร้างข้อมูลที่เป็นกลุ่มจะมองแบบสตรีม ซึ่งหมายถึงเป็นแบบ linear line terminate with zero หรือแบบ text ที่เป็น 4 บรรทัด แต่ละบรรทัดปิดท้ายด้วย \null ดังนั้น โครงสร้าง text จะเป็นแบบตัวชี้ 4 ชุด ชี้ไปที่อะเรย์ 4 ชุด เช่นกัน

ในลักษณะของอุปกรณ์อื่น ๆ เช่น คีย์บอร์ด เครื่องพิมพ์ก็เช่นเดียวกัน แต่บางหมายเลขนั้นจะมีลักษณะเฉพาะไปตามอุปกรณ์

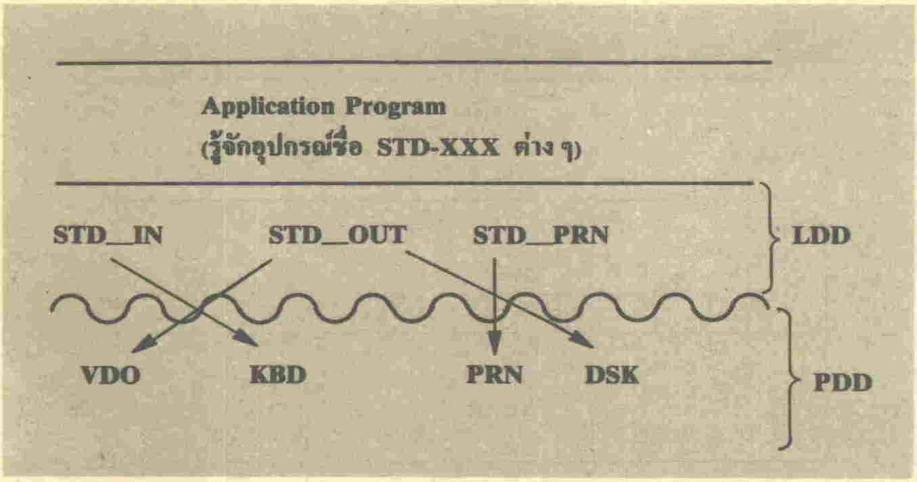
ลอจิคอลดีไวซ์ไดรเวอร์ (LDD-Logical Device Driver)

มีลักษณะเป็นการกำหนดหมายเลขหรือชื่อให้กับอุปกรณ์ โดยที่มีชื่ออุปกรณ์ทางลอจิกอยู่แล้ว เช่น STD_IN, STD_OUT, STD_ERR, STD_PRN, STD_AUX

การกำหนดหมายเลขหรือชื่อให้กับอุปกรณ์ทางด้านลอจิกนี้ก็เพื่อประโยชน์ในการกำหนดทิศทาง (redirection) ของอุปกรณ์เหล่านั้นได้ โครงสร้าง LDD เมื่อกำหนดทิศทางใหม่เข้าสู่ PDD แสดงดังรูปที่ 2

หมายเลข	ชื่อฟังก์ชัน
0	– Set up VDO system
1	– get VDO system
.	.
.	.
.	.
16	– Display character at specific position with attribute
17	– Display character at specific position
18	– Display character at cursor position with attribute
19	– Display character at cursor position
.	.
.	.
22	– Display stream at specific position with attribute
25	– Display text at specific position with attribute.

ตารางที่ 1 ตัวอย่างฟังก์ชันบางหมายเลขที่เรียกเข้าหาระบบ



รูปที่ 2 โครงสร้าง LDD เมื่อกำหนดทิศทางเข้าสู่ PDD

ฟังก์ชันคอลล์และไลบรารี่ฟังก์ชัน (FCLF-Function Call and Library Function)

ในระดับนี้จะมีอยู่ 2 ระบบคือระบบที่เกี่ยวกับอินพุตเอาต์พุต (I/O System) และระบบเกี่ยวกับภาษาไทย (Thai system)

ระบบที่เกี่ยวกับอินพุตเอาต์พุต ได้แก่ระบบที่ใช้ในการติดต่อกับอินพุตหรือเอาต์พุต โดยจะติดต่อเข้าทาง PDD ก็ได้ ลักษณะการติดต่อจะมีได้ 2 แบบ คือเป็นหมายเลขฟังก์ชัน (function number) หรือเรียกโดยฟังก์ชันในไลบรารี่ (library function) กล่าวคือเป็นฟังก์ชันที่เขียนขึ้นเก็บไว้ในไลบรารี่ เพื่อให้ผู้เขียนซอฟต์แวร์ประยุกต์เรียกใช้ได้ง่าย

ระบบเกี่ยวกับภาษาไทย เป็นระบบที่ทำการจัดการทางด้านภาษาไทย เช่น การค้นหา การจัดเรียง การจัดรูปแบบข้อความภาษาไทย ฯลฯ ระบบจะได้รับการเรียกใช้งานจากโปรแกรมประยุกต์ ระบบนี้จะมีลักษณะการติดต่อเรียกใช้เป็นแบบฟังก์ชันไลบรารี่ (library function)

โปรแกรมประยุกต์

เป็นส่วนที่เขียนขึ้นเพื่อประยุกต์ใช้งานตามความต้องการของผู้เขียน โดยสามารถเรียกการบริการของระบบ TKS ได้ในทุกระดับ ตั้งแต่ FCLF, LDD ลงไปถึง PDD การเรียกนี้ขึ้นอยู่กับว่าต้องการความรวดเร็วหรือความสะดวกเพียงไร การ

เรียกลงระดับ PDD ก็จะทำให้โปรแกรมทำงานได้เร็วขึ้น แต่ก็สามารถเรียกในระดับ FCLF ได้ ซึ่งจะทำให้ผู้เขียนเขียนโปรแกรมได้ง่ายและรวดเร็วขึ้น

โครงสร้างข้อมูลภาษาไทยใน TKS

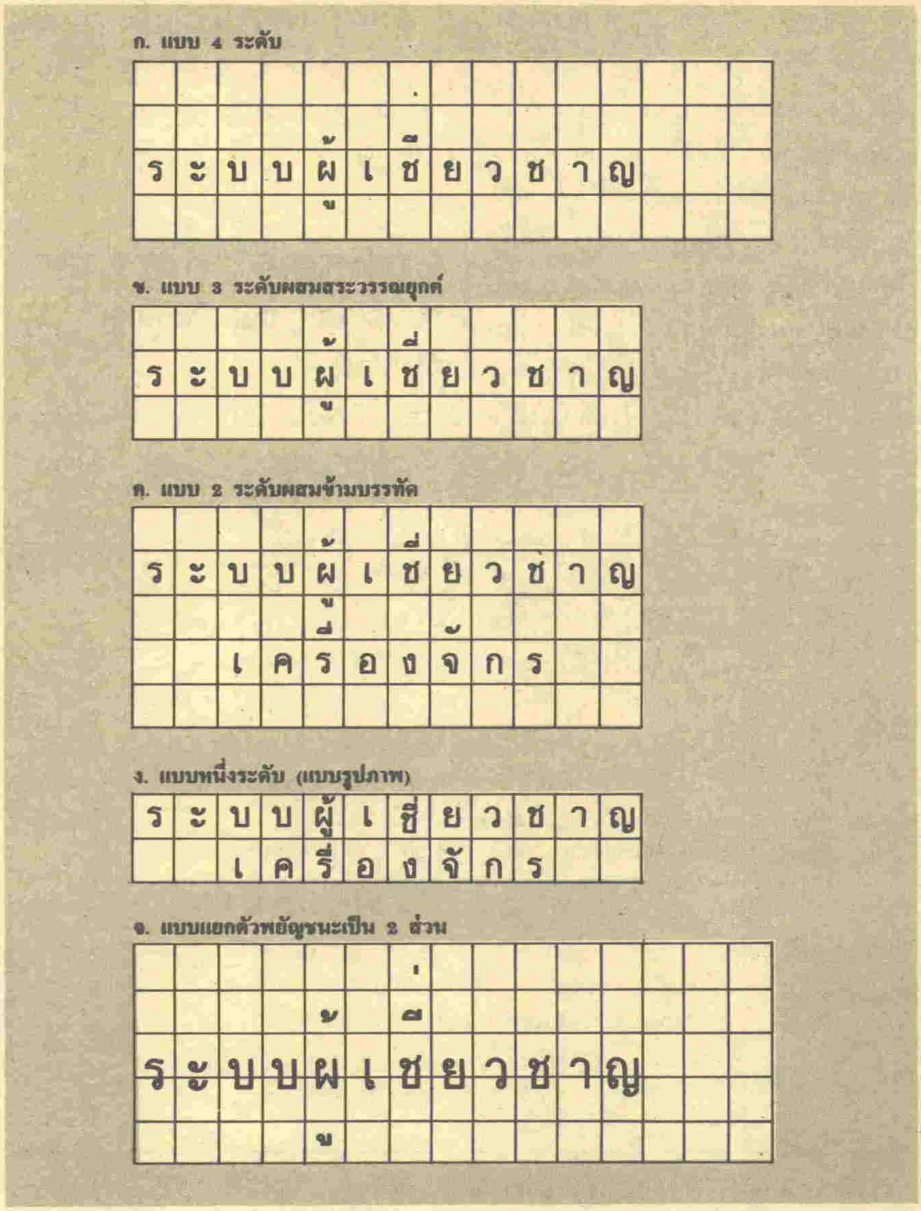
สืบเนื่องจากโครงสร้างการประยุกต์ภาษาไทยบนไมโครคอมพิวเตอร์มีการจัดรูปแบบของการแสดงผลข้อมูลได้หลายแบบ แต่ละแบบก็มีลักษณะเฉพาะตัว ผู้ประยุกต์หรือสร้างไทยการ์ดจะพยายามยึดหลักในแนวทางที่แตกต่างกันออกไป โดยปกติแล้วทุกแบบจะมีทั้งข้อดีและข้อเสียของการแสดงผล

รูปแบบของการแสดงผลในแง่ของฮาร์ดแวร์หรือการจัดระดับเพื่อแสดงผลเป็นได้ดังรูปที่ 3

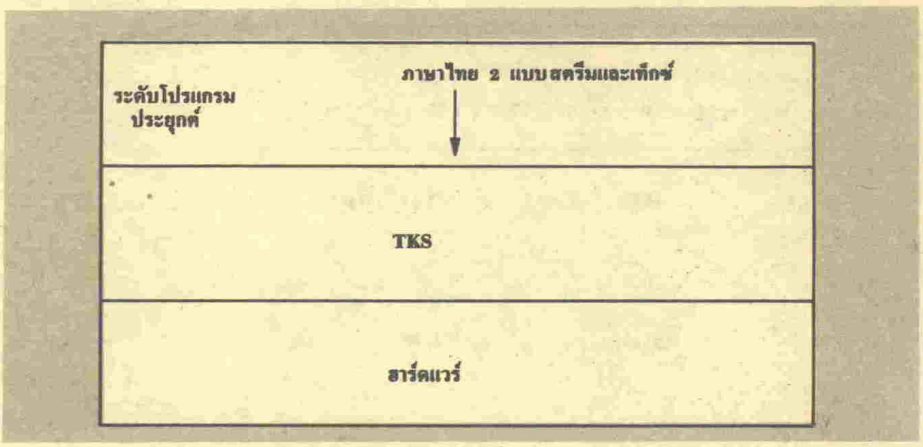
ด้วยวิธีการแสดงผลภาษาไทยหลายแบบนี้จะขึ้นกับความเหมาะสมของฮาร์ดแวร์แต่ละแบบ เช่นแบบหนึ่งระดับเหมาะที่จะแทนตัวอักษรเป็นแบบรูปภาพและกำหนดรหัสเป็นสองไบต์ แสดงผลแบบกราฟิกได้ แบบที่แสดงผลในจอภาพของเครื่องไมโครคอมพิวเตอร์ 16 บิตที่แสดงผลภาษาไทย 25 บรรทัดแบบอินเตอร์เลสโหมด มักนิยมการแสดงผลในแบบสองระดับผสมข้ามบรรทัด ส่วนแบบแยกตัวพยัญชนะเป็นสองส่วนนี้โดยทั่วไปจะมีวงจรทางฮาร์ดแวร์เป็นพิเศษ

ภาษาไทยที่ใช้กับ TKS ในระดับโปรแกรมประยุกต์จะเหลือ 2 แบบ

เพื่อการกำหนดให้ระดับโปรแกรมประยุกต์เป็นมาตรฐานใช้งานร่วมกันได้ ดังนั้นจำเป็นจะต้องนิยามภาษาไทยให้เหลือรูปแบบน้อยที่สุดที่จะใช้งานได้อย่างมีประสิทธิภาพ



รูปที่ 3 โครงสร้างการจัดแสดงภาษาไทยขึ้นกับฮาร์ดแวร์ซึ่งจัดแสดงได้หลายแบบ



รูปที่ 4 ไดอะแกรมแสดงให้เห็นการเชื่อมในระดับโปรแกรมประยุกต์

ดังนั้น TKS จึงต้องพยายามสร้างลักษณะโครงสร้างของตัวอักษรภาษาไทยที่สามารถแทนได้ทุกกรณีไม่ว่าจะใช้วิธีใด TKS จะมองตัวอักษรภาษาไทยออกเป็นสตรีม (stream) และเท็กซ์ (text) เท่านั้น

โครงสร้างสตรีม เป็นชุดของตัวอักษรและอักขระควบคุมเรียงกันไปและปิดท้ายด้วย zero เช่นในรูปที่ 5

โดยที่อักขระแต่ละตัวนั้นจะเป็นไบต์หรือเวิร์ดขึ้นอยู่กับโครงสร้างของรหัสว่าจะเป็นแบบ one byte หรือ two byte code

ลักษณะของข้อมูลเป็น

typedef STREAM (อักขระ*);

ส่วนโครงสร้างของ TEXT เป็นชุดข้อมูลของอักขระและอักขระควบคุมในการจัดแบบ 4 ระดับ ทุกระดับปิดท้ายด้วย zero เช่นกัน ดังในรูปที่ 6

โดยที่ 1 cell ของข้อมูลจะแทนด้วย 1 ไบต์ หรือเวิร์ดก็ได้ ลักษณะของข้อมูลจะเป็นดังนี้

word ก็ได้ ลักษณะของข้อมูลจะเป็นดังนี้

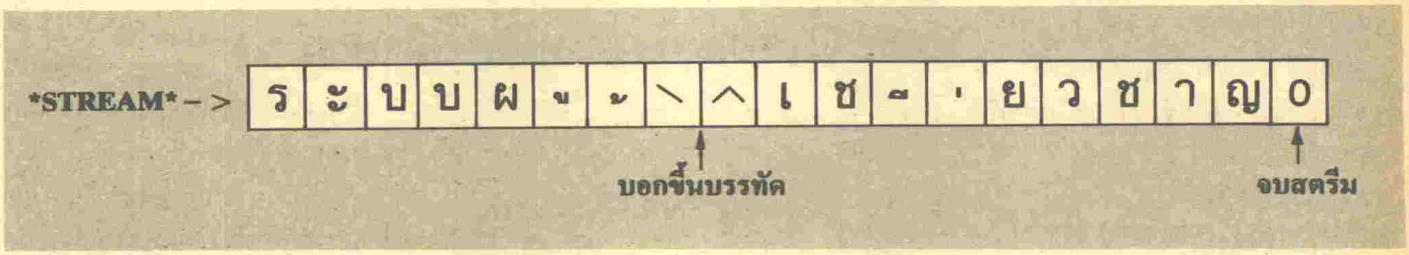
```
typedef TEXT struct {
        อักขระ *VANA, *SALA,
        *PYAN, *SALA_L
        } (*);
```

ด้วยโครงสร้าง 2 แบบนี้จะเป็นโครงสร้างที่โปรแกรมประยุกต์มองเห็น หรือเรียกใช้ผ่าน TKS ส่วนใน TKS จะจัดโครงสร้างการแสดงผลอย่างไรก็จะขึ้นกับสิ่งแวดล้อมของอุปกรณ์ หรือไทยการ์ดที่ใช้ตามความเหมาะสม ด้วยวิธีการกำหนดดังที่กล่าวนี้จะทำให้รูปแบบการเรียกใช้ภาษาไทยของซอฟต์แวร์ประยุกต์มีมาตรฐานที่ไม่สับสนกับโครงสร้างทางฮาร์ดแวร์อีกต่อไป

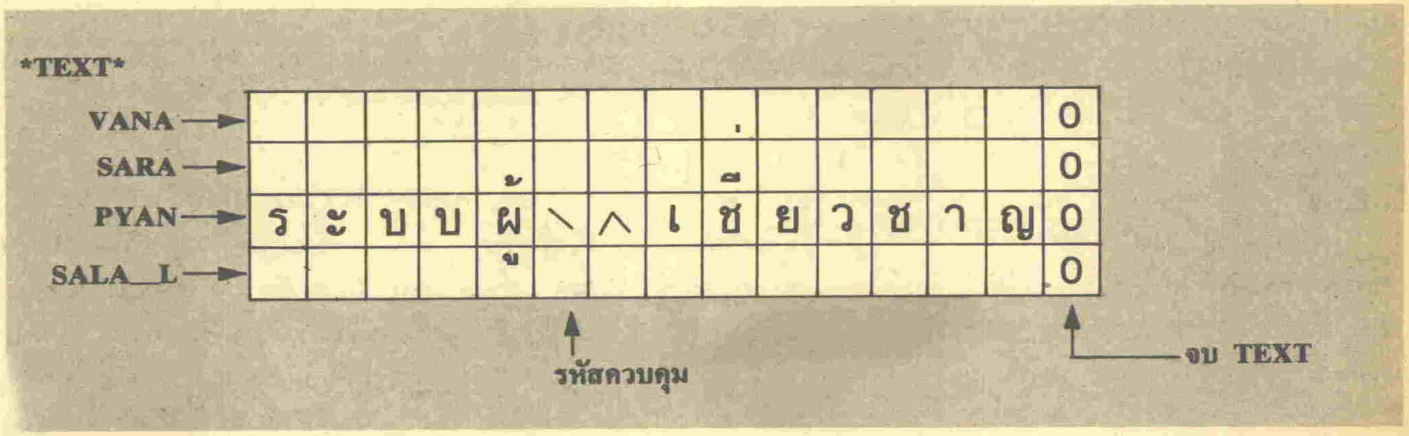
TKS จะอยู่ร่วมในระบบอย่างไร

ในการต่อเชื่อม TKS เข้าในระบบเรามีวิธีการนำ TKS เข้าในระบบหลายรูปแบบตามโครงสร้างของโอเอสเช่น ถ้าเป็นเมนเฟรมคอมพิวเตอร์หรือมินิคอมพิวเตอร์เราจะถือว่า TKS เป็นอิสระโดยโปรแกรมประยุกต์จะผ่านชั้นของ TKS ที่เราสร้างขึ้น ดังนั้น TKS ในลักษณะนี้จะมีการประยุกต์ให้ฟังก์ชันเป็นมาตรฐาน การเรียกเข้าในระบบจึงไม่เป็นปัญหา การเขียนโปรแกรม TKS ก็จะเขียนด้วยภาษาระดับสูงได้

สำหรับกรณีของไมโครคอมพิวเตอร์ TKS ในส่วนของฟิสิคัลดีไวซ์ไดรเวอร์จะแทรกตัวร่วมกับไบออสฟังก์ชันและมีสวิตช์เลือกเข้าออก เพื่อให้การเรียกใช้ไบออสผ่าน TKS หรือไม่ผ่าน ส่วนในระดับลอจิคอลดีไวซ์ไดรเวอร์จะทำงานร่วมกับเอ็มเอสดอสเพื่อกำหนดทิศทางการเข้าหาระบบได้ถูกต้อง ส่วนฟังก์ชันคอลล์จะเป็นโมดูลแยกอิสระลอยอยู่เหนือโอเอส โครงสร้างในการทำงานร่วมกับโอเอสและไบออส จะได้มีการกล่าวในรายละเอียดต่อไปในตอนหน้า



รูปที่ 5 โครงสร้างของสตรีม



รูปที่ 6 โครงสร้างของแท็กซ์

TKS เขียนด้วยภาษาอะไร

หากมีการกำหนดสเปคเป็นมาตรฐานของ TKS แล้ว ผู้พัฒนาฮาร์ดแวร์หรือระบบไทยการ์ดก็จะเขียนโมดูลในส่วนของ PDD ตามฟังก์ชันและพารามิเตอร์ที่กำหนด เมื่อเป็นเช่นนี้โปรแกรมในระดับสูงกว่าจะเรียกใช้ได้โดยไม่มีปัญหาและจะพบว่าไม่ขึ้นกับฮาร์ดแวร์และรหัสที่เป็นปัญหาอีกต่อไป

สำหรับ TKS ที่จะแนะนำในตอนนี้เขียนขึ้นด้วยภาษาแอสเซมบลีและจัดการกับจอภาพได้ทั้ง CGA และโมโนโครมในโหมดกราฟิก กล่าวคือตัวอักษรภาษาไทยที่แสดงบนจอภาพจะไม่ใช้ ROM กำเนิดตัวอักษร แต่จะใช้โปรแกรมสร้างภาพตัวอักษร เนื่องจากมีการวางโครงสร้างต่าง ๆ ไว้อย่างดี ระดับความเร็วในการแสดงผลเมื่อผู้เขียนโปรแกรมเรียกผ่าน TKS จึงเร็วพอที่จะใช้งานได้โดยไม่มีปัญหา ในฉบับนี้ได้ให้ซอฟต์เมล์ของโปรแกรม TKS ไว้ แต่เนื่องจากฟังก์ชันของ TKS มีอยู่มากในตอนนี้จะยังไม่ขออธิบายวิธีการเรียกเข้าหาระบบครบทุกฟังก์ชัน แต่จะให้พอเป็นแนวความคิดก่อน

ตัวอย่างการเรียกใช้ TKS

ระบบ TKS ที่ออกแบบและสร้างขึ้นนี้เน้นหลักการให้เรียกใช้ในระดับซอฟต์แวร์ประยุกต์ได้ง่าย ในที่นี้ขอยกตัวอย่างการเรียกใช้ TKS ในแบบต่าง ๆ กันดังนี้

ตัวอย่างที่ 1 การเรียกใช้ผ่าน LDD เพื่อส่งข้อความออกแสดงผลทาง STD_OUT ซึ่งทำหน้าที่เป็น device print stream

```
main ( )
    {
    STRING string;
    string = & "ระบบ ผ ๆ ๆ\ ^ เ ช ๔ ๑ ย ว ช า ญ \ 0";
    dprints (STD_OUT, string);
    exit ( );
    }
```

หมายเหตุ ฟังก์ชัน dprints เป็นฟังก์ชันในไลบรารี่ (FCLF)

ตัวอย่างที่ 2 มีลักษณะคล้ายกับตัวอย่างที่ 1 แต่เป็น device print text คือเปลี่ยนจาก stream เป็น text

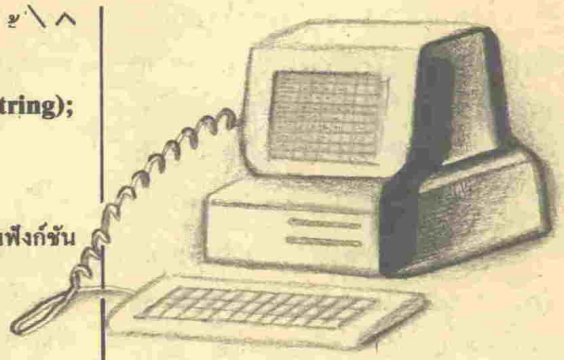
```
main ( )
    }
    STREAM str;
    TEST txt;
    str = &'ระบบ ผ ๆ ๆ\ ^ เ ช ๔ ๑ ย ว ช า ญ \ 0";
    txt = STM_TXT (str); /*convert stream to text*/
    dprintt (STD_PRN, txt);
    exit ( );
    }
```

หมายเหตุ ฟังก์ชัน dprintt เป็นฟังก์ชันในไลบรารี่ (FCLF)

ตัวอย่างที่ 3 เป็นตัวอย่างการเรียกเข้าหาระบบ (function call) ตามหมายเลขฟังก์ชันที่กำหนด เพื่อทำการอ่านตัวอักษรจาก STD_AUX

```
= define READCHAR 5 /* Function call number */
main ( )
    {
        Char ch;
        ch = Func _call (READ-CHAR, STD_AUX);
        exit ( ) ;
    }
```

หมายเหตุ ฟังก์ชัน Func_call เป็นฟังก์ชันอยู่ใน FCLF



ตัวอย่างที่ 4 เป็นตัวอย่างการเรียกเข้าหา PDD โดยตรง เพื่อส่งข้อความ text ไปแสดงผล

```
= define ENTRY 17
main ( )
{
TEXT text = {"_", "_", "_", "_"};
PDD_VDO (ENTRY 17, txt);
}
```

หมายเหตุ PDD_VDD เป็นชื่อการเรียกเข้าหาระบบในระดับ PDD

ส่งท้าย

ในการใช้งาน TKS นี้ ผู้ใช้เพียงแต่เขียนโปรแกรมและเรียกเข้าหาระบบ ระบบ TKS จะจัดการเกี่ยวกับตัวอักษรที่แสดงผลและออกเอาต์พุตหรือรับอินพุตจากคีย์บอร์ดให้อย่างถูกต้องตามอักขระวิธี ในฉบับนี้จะเน้นให้เห็นแนวความคิดของ TKS ก่อน ในฉบับต่อ ๆ ไปจะได้นำรายละเอียดของการประยุกต์และตัวอย่างการเรียกใช้มาเสนอต่อไป

เอกสารอ้างอิง

1. ยืน ภู่วรวรรณ และคณะ **"ระบบภาษาไทยระดับแก่น ข้อเสนอแนะเพื่อจัดทำเป็นมาตรฐาน"** เอกสารประกอบการประชุมสัมมนาวิชาการวิศวกรรมไฟฟ้า 9 สถาบันอุดมศึกษาครั้งที่ 10, 17 ธ.ค. 2531